

# SCC2692 differences from the SCN2681

AN414

Author: D. Ibarra

## DESCRIPTION

The SCC2692 is a CMOS version of the SCN2681 DUART. The SCC2692 is functionally and pin-to-pin compatible with the SCN2681 and can be substituted into existing SCN2681 designs. There are a few additional features and differences in the SCC2692 which are discussed below.

The major difference between the two parts is that the SCC2692 is CMOS, while the SCN2681 is NMOS. This means the SCC2692 draws significantly less power than the SCN2681.

Another difference between the two parts is that the SCC2692 has used edge triggered latches for the configuration registers: MR1 MR2 CSR. OPCR and ACR. The SCN2681 uses transparent latches, gated by WRN AND CEN. So, because of the undefined state of the data bus prior to the data being valid, both the receiver and transmitter must be disabled before writing to any register which could affect them, even if the same data is being rewritten. The SCC2692 will not be affected by the undefined data and will allow, for example the transmitter configuration to be changed without affecting the receiver operation. Only the transmitter would have to be disabled for the following: changing the transmitted stop bit length; changing the value of the transmitted ninth bit during wake-up mode; and changing the clock source or value for the transmitter only. It is still recommended to disable the transmitter or receiver before changing their respective configurations.

## SOFTWARE

The SCN2681 counter/timer will be in timer mode with the timer running after power up and after a hardware reset. The SCC2692 counter/timer will also be in timer mode after power up and after a hardware reset, but the timer will not be running until it receives a start counter/timer command (read at address H'E'). After this initial start the SCC2692 timer will run continuously. Subsequent start C/T commands will cause either the SCC2692 or the SCN2681 to immediately load the values in the CTUR and CTLR and start a new cycle. The stop counter command will reset the counter ready status bit (ISR[3]), but will have no effect on the timer mode operation for either part.

Utilizing command register (CR) bit 7 which is an unused bit in the SCN2681 makes six additional commands available to the user. SCN2681 programs which set this bit to zero will not be affected by this change and will run with the SCC2692.

## Timeout Mode

This mode is useful for when the user programs RxRDY to interrupt the CPU when the receive FIFO is full. When less than three characters are received, the FIFO won't become full and the CPU won't get an RxRDY interrupt. The timeout mode provides the user with a time out interrupt via the counter/timer. If characters are received and the FIFO does not become full, a preselected period of delay can be timed out by the counter/timer (C/T) and the CPU interrupted.

The counter/timer is used in this mode by programming CTUR/CTLR with a value greater than the normal receive character period. Each time a received character is transferred from the shift register to the receive holding register (RHR), the C/T is reloaded with the value in CTLR/CTUR and then restarted. If a continuous data stream is not received and the C/T is allowed to end the count, the counter ready bit ISR[3], will be set. If the interrupt mask, IMR[3], has been set, an interrupt will be generated. This mode is

enabled by writing CR = H'A0', and is turned off by writing CR = H'CO'.

## Power Down Mode

Power down mode can be used for energy conservation during idle periods. This mode saves the contents of all the internal registers, stops the oscillator and suspends the operation of any function that uses the oscillator. In addition, the current used by the part is reduced. The part can be put into power down mode at any time and restored to normal operation when needed. Since all register values are saved, re-initialization is not necessary.

To put the part into power down mode:

- CRA = H'30' — Reset Tx A
- CRA = H'20' — Reset Rx A
- CRB = H'30' — Reset Tx B
- CRB = H'20' — Reset Rx B
- CRA = H'E0' — power down mode on

To get out of power down mode:

- CRA = H'F0' — Power down mode off
- Wait for oscillator to start up
- CRA = H'05' — Enable Rx A and Tx A
- CRB = H'05' — Enable Rx B and Tx B

## Set/Reset RTSN Output

The RTSN output (OP0 and OP1) can be programmed to be asserted or negated automatically by either the receiver or the transmitter and it can be manually asserted and negated. Both the SCN2681 and the SCC2692 provide the ability to manually set and reset the output port bits by writing the appropriate bit mask to the Set output port bits command address or to the Reset output port bits command address (see data sheets for details). The SCC2692 as an additional feature provides commands in the CR to easily assert and negate just the RTSN output. CR H'80' will assert RTSN, cause the output to go low while CR = H'90' will negate RTSN, cause the output to go high.

## HARDWARE

The on board oscillator circuitry in the SCC2692 is different from the SCN2681. This was required because of the addition of the power down mode logic. SCN2681 boards which use a crystal between the X1 and X2 pins with an equal value capacitor from X1 to ground as from X2 to ground can be used unmodified with the SCC2692. The SCC2692 does not require an external resistor but will operate with one there. SCN2681 boards which use an external oscillator to drive a signal into X1 and have X2 grounded can be used with the SCC2692.

The oscillators on both parts are the same basic type, consisting of an inverter and feedback resistor between the X1 and X2 pins, which, with the addition of an external crystal and capacitors, are used to implement a Pierce oscillator. The SCN2681 has the output of the inverter connected to X1, the input of the inverter connected to X2, and the inverter is a Schmitt trigger. The SCC2692 has the input of the inverter connected to X1 and the output of the inverter connected to X2.

Because of the Schmitt trigger inverter, the SCN2681 is limited to using small value external capacitors. If capacitors of 15pF or greater are used, intermittent power-on problems may be experienced. The oscillator may stay in relaxation mode oscillating at a frequency much lower than the one the crystal is specified for. For this reason we recommend using external capacitors of around

# SCC2692 differences from the SCN2681

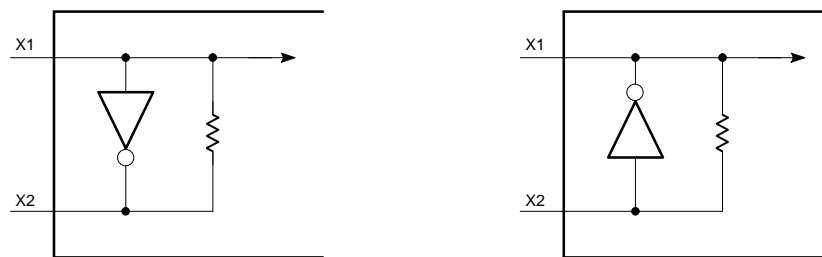
# AN414

5pF and board supplied stray capacitance of no more than 5pF. It has also been found that adding an external resistor of 100k–1 MΩ across X1 and X2 will solve other start up problems for some designs. While we recommend using balanced capacitors (C1=C2) of 5pF, unbalanced values may be used. Since the X2 pin (input to the internal inverter) has the most sensitivity to capacitance, many designs use a larger value capacitor (10–15pF) on the X1 pin while leaving 5pF on the X2 pin.

The SCC2692 oscillator is not limited to the recommendations required for the SCN2681. It does not need an external resistor and will operate with a much wider range of external capacitor values. New designs using the SCC2692 oscillator should use a parallel calibrated crystal and the external capacitor values should be adjusted until the total circuit capacitance matches the capacitance

specified for the crystal. The capacitor values are not limited to any specific value but the two capacitors should be approximately equal in value. Using two 24pF capacitors and a 3.6864MHz crystal with CL = 20pF will give accurate reliable results. A source for this crystal is: Saronix, Palo Alto, CA, part number NYP037–20. From California call (800) 422-3355; outside California call (800) 227-8974.

Some designs will not use the on board oscillator circuitry but will drive the part with an external clock source. The SCN2681 and the SCC2692 both require that the external clock be driven into the X1 pin. The SCN2681 requires that the X2 pin be grounded while the SCC2692 X2 pin can be grounded or left open. Alternately, the X2 input can be driven with the complement of the clock signal going to X1 for both parts.



Oscillator Configuration